

תרגיל מס' 1: חיפוש בمبرור – וIMPLEMENTATION_ID

מצורפים הקבצים הבאים:

stack.py – מימוש מבנה הנתונים: מחסנית.

frontier.py – מימוש מבנה נתונים המתאים ל-ID: מחסנית "משוכללת" – מחסנית שמאחסנת

מצבים עד עומק חיפוש D ; כאשר היא מתפרקת היא תוענת שוב את המצב ההתחלתי ומגדילה את D ב-1.

state.py – מימוש מחלוקת של Maze בגודל $3 \times n$.

Search.py – מימוש של החיפוש, כפי שנלמד בהרצאה.

עליכם לעبور על הקבצים המצורפים, להבין אותם ולבצע את המשימות הבאות:

1. שנו את מבנה הנתונים כך שישמור את מספר המצבים המאוחסנים כרגע במבנה הנתונים.

כתוצאה מזה, יכול 5 פריטים ולא 4:

[stack, max. depth, initial. state, try next level?, total items pushed]

2. צרו **frontierStack** שימממש מחסנית, וישמר את הפרטים הבאים:

[Stack, total item pushed]

3. הגדרו מחלוקת **Queue** שתממש התנאיות של תור תור שימוש ברשימה.

צרו **frontierQueue** שימממש תור, וישמר את הפרטים הבאים:

[Queue, total item pushed]

4. כתבו תוכנה שתירץ 100 הריצות של בניית מבוכים בגודל 5^5 ומציאת פתרון עבורי, בשיטת DFS,

Iterative Deepening, BFS, ו-DFS.

מוצג בדיקות שנדרכו עבור מבוכים שנתקעו (שלא היה להם פתרון), עד שהאלגוריתם עצר והודיע שאין פתרון	אורך מקסימלי של מספר המצבים שהי בתוך התור בו זמןית (מוצג עבור המבוכים שהצליחו להגיע אל היעד)	מספר בדיקות מוצג (מוצג עבור המבוכים שהצליחו להגיע לייעד).	אחוז המבוכים שנמצא בהם מסלול מהמקור לייעד	
				DFS
				BFS
				Iterative Deepening

5. מה מסקנתכם?

בצלחה רבה!